

Package ‘landsepi’

October 5, 2019

Type Package

Encoding UTF-8

Title Landscape Epidemiology and Evolution

Version 0.0.8

Date 2019-10-04

Author Loup Rimbaud [aut],
Julien Papaix [aut],
Jean-Francois Rey [cre]

Maintainer Jean-Francois Rey <jean-francois.rey@inra.fr>

Description A spatio-temporal stochastic model to assess resistance deployment strategies against plant pathogens. The model is based on stochastic geometry for describing the landscape and the resistant hosts, a dispersal kernel for the dissemination of the pathogen, and a SEIR (Susceptible-Exposed-Infectious-Removed) architecture to simulate plant response to disease.

Loup Rimbaud, Julien Papaix, Jean-François Rey, Luke G Barrett,
Peter H Thrall (2018) <doi:10.1371/journal.pcbi.1006067>.

URL <https://gitlab.paca.inra.fr/CSIRO-INRA/landsepi>

BugReports <https://gitlab.paca.inra.fr/CSIRO-INRA/landsepi/issues>

License GPL (>= 2) | file LICENSE

LazyData true

BuildVignettes true

NeedsCompilation yes

Biarch true

SystemRequirements C++11, gsl, gdal >= 1.11.0

Depends methods, utils, grDevices (>= 3.0.0), graphics (>= 3.0.0), R (>= 3.3.0), Rcpp (>= 0.9.0), rgdal (>= 1.2-16)

Imports sp (>= 1.0-17), stats (>= 3.0.2), Matrix, MASS, rgeos, mapproj, fields, splancs, sf

Collate 'invlogit.R' 'logit.R' 'RcppExports.R' 'graphLand.R'
 'periodic_cov.R' 'multiN.R' 'AgriLand.R' 'HLIRdynamics.R'
 'demo_landsepi.R' 'landsepi.R' 'plotevolQR.R'
 'simul_landsepi.R'

LinkingTo Rcpp

RoxygenNote 6.1.1

Repository CRAN

Date/Publication 2019-10-04 23:30:02 UTC

R topics documented:

landsepi-package	2
AgriLand	5
demo_landsepi	7
dispH	8
dispP_1	8
dispP_2	9
dispP_3	10
dispP_4	11
dispP_5	11
HLIRdynamics	12
invlogit	14
landscapeTEST1	15
landscapeTEST2	16
landscapeTEST3	16
landscapeTEST4	17
landscapeTEST5	18
logit	18
model_landsepi	19
multiN	21
periodic_cov	22
plotevolQR	23
plotland	23
simul_landsepi	24

Index **29**

landsepi-package *Landscape Epidemiology and Evolution*

Description

A spatio-temporal stochastic model to assess resistance deployment strategies against plant pathogens. The model is based on stochastic geometry for describing the landscape and the resistant hosts, a dispersal kernel for the dissemination of the pathogen, and a SEIR (Susceptible-Exposed-Infectious-Removed) architecture to simulate plant response to disease.

Details

Package: landsepi
Type: Package
Version: 0.0.8
Date: 2019-10-04
License: GPL (>=2)

The landsepi package implements a spatially explicit stochastic model able to assess the epidemiological and evolutionary outcomes of four major strategies to deploy plant resistance to pathogens. These strategies include the combination of several resistance sources across time (crop rotations) or space. The spatial scale of deployment can vary from multiple resistance sources occurring in a single cultivar (pyramiding), in different cultivars within the same field (cultivar mixtures) or in different fields (mosaics). The simulated sources of resistance can consist of qualitative resistance (i.e. major genes) or quantitative resistance traits (including Adult Plant Resistant genes) against several components of pathogen aggressiveness: infection rate, latent period duration, propagule production rate, and infectious period duration. This model provides a useful tool to assess the performance of a wide range of deployment options, and helps investigate the effect of landscape, epidemiological and evolutionary parameters on the performance of a given strategy.

The simulation model is based on a SEIR (Susceptible-Exposed-Infectious-Removed) architecture to describe host response to disease. The landscape is represented by a set of polygons where the pathogen can disperse. Initially, the pathogen is not adapted to any source of resistance, and is only present on susceptible hosts. However, through mutation, it can evolve and may acquire infectivity genes (which leads to breakdown of major resistance genes) or increase aggressiveness (which leads to the erosion of the relevant quantitative resistance traits). Furthermore, loci may be re-assorted via sexual reproduction. However, evolution of a pathogen toward infectivity or increased aggressiveness on a resistant host may be penalised by a fitness cost on susceptible hosts. Consequently, pathogens carrying infectivity genes may have reduced infection rate (cost of infectivity) on susceptible hosts relative to pathogens that do not carry these genes. Similarly, a gain in pathogen aggressiveness on quantitatively resistant hosts is penalised by a decreased aggressiveness on susceptible hosts, leading to a trade-off.

The evolutionary outcome of a deployment strategy is assessed by measuring the time until the pathogen reaches the three steps to adapt to plant resistance:

- (d1) first appearance of adapted mutants,
- (d2) initial migration to resistant hosts and infection, and
- (d3) broader establishment in the resistant host population (i.e. the point at which extinction becomes unlikely).

Epidemiological outcomes are evaluated using:

- (e1) the Green Leaf Area (GLA) as a proxy for yield, and
- (e2) the area under the disease progress curve (AUDPC) to measure disease severity.

The package includes five examples of landscape structures and a default parameterisation to represent plant pathogens as typified by rusts of cereal crops (genus *Puccinia*, e.g. stripe rust, stem rust and leaf rust of wheat and barley). The main function of the package is `simul_landsepi()`. It can

be parameterised to simulate various resistance deployment strategies using the provided landscapes and parameters for cereal rusts.

A set of graphics and a video showing epidemic maps can also be generated.

Future versions:

Future versions of the package will include in particular:

- A more flexible parameterisation of pathogen life-history traits, in order to simulate other plant pathogens.
- A more flexible parameterisation of deployment strategies, in order to simulate complex strategies combining several options (e.g. mosaic of pyramids) as well as the allocation of more than 3 different cultivars in the landscape.

Dependencies:

The package for compiling needs:

- g++
- libgs2
- libgs-dev
- gdal-bin
- libgdal-dev

and the following R packages:

- Rcpp
- rgdal
- sp
- stats
- Matrix
- MASS
- rgeos
- maptools
- fields
- splancs
- sf

In addition, to generate videos the package will need ffmpeg.

Author(s)

Loup Rimbaud <loup.rimbaud@inra.fr>

Julien Papaix <julien.papaix@inra.fr>

Jean-Francois Rey <jean-francois.rey@inra.fr>

Maintainer: Jean-Francois Rey <jean-francois.rey@inra.fr>

References

When referencing the simulation model, please cite the following article:

Rimbaud L., Papaix J., Rey J.-F., Barrett L. G. and Thrall P. H. (2018). Assessing the durability and efficiency of landscape-based strategies to deploy plant resistance to pathogens. *PLoS Computational Biology* 14(4):e1006067.

When referencing the R package, please cite the following package:

Rimbaud L., Papaix J. and Rey J.-F. (2018). landsepi: Landscape Epidemiology and Evolution. *R package*, url: <https://cran.r-project.org/package=landsepi>.

See Also

Useful links:

- <https://gitlab.paca.inra.fr/CSIRO-INRA/landsepi>
- Report bugs at <https://gitlab.paca.inra.fr/CSIRO-INRA/landsepi/issues>

Examples

```
## Not run:
library("landsepi")

## Run a demonstration (a 30-year simulation of a mosaic deployment strategy of two
## resistant cultivars in balanced proportions and high level of spatial aggregation)
demo_landsepi()

## Run a simulation with data included in the package (default parameterisation:
## 5-year simulation of a mosaic deployment strategy of two resistant cultivars
## in balanced proportions and high level of spatial aggregation)
simul_landsepi()

## (see ?simul_landsepi to help parameterise the function and simulate other scenarios)

## End(Not run)
```

AgriLand

Landscape generation

Description

Generate a landscape composed of fields where a susceptible (SC) and one (RC) or two (RC1 and RC2) resistant cultivars are allocated with controlled proportions and spatio-temporal aggregation.

Usage

```
AgriLand(landscape, filename = "landscapeTEST1", propSR, iso1SR, propRR,
         iso1RR, strat, Nhost, nYears, seed, graphic = FALSE)
```

Arguments

landscape	a spatialpolygon object containing field coordinates.
filename	a character string specifying the output layer name.
propSR	proportion of fields where resistance is deployed: $(RC)/(SC+RC)$ or $(RC1+RC2)/(SC+RC1+RC2)$. Must be between 0 and 1.
isolSR	an integer giving the spatial aggregation of fields where resistance is deployed (1=highly fragmented, 2=balanced, 3=highly aggregated).
propRR	when applicable (mixtures and mosaics only), relative proportion of the second resistant cultivar: $(RC2)/(RC1+RC2)$. Must be between 0 and 1.
isolRR	when applicable, an integer giving the spatial (for mosaics) or temporal (for rotations) aggregation of fields cultivated with the second resistant cultivar (1=highly fragmented, 2=balanced, 3=highly aggregated).
strat	a character string specifying the deployment strategy ("MO"=mosaic, "MI"=mixture, "RO"=rotations, "PY"=pyramiding).
Nhost	an integer giving the number of cultivars (1, 2 or 3).
nYears	an integer giving the number of simulated years.
seed	an integer giving the seed value (for random number generator).
graphic	a logical indicating if a graph of the landscape must be generated (TRUE) or not (FALSE).

Details

An algorithm based on latent Gaussian fields is used to allocate two different crop cultivars across the simulated landscapes (e.g. a susceptible and a resistant cultivar, denoted as SC and RC, respectively). This algorithm allows the control of the proportions of each cultivar in terms of surface coverage, and their level of spatial aggregation. A random vector of values is drawn from a multivariate normal distribution with expectation 0 and a variance-covariance matrix which depends on the pairwise distances between the centroids of the fields. Next, the crop cultivars are allocated to different fields depending on whether each value drawn from the multivariate normal distribution is above or below a threshold. The proportion of each cultivar in the landscape is controlled by the value of this threshold. The sequential use of this algorithm allows the allocation of more than two crop cultivars (e.g. SC, RC1 and RC2). Therefore, deployment strategies involving two sources of resistance is simulated by:

1. running the allocation algorithm once to segregate the fields where the susceptible cultivar is grown, and
2. applying one of the following deployment strategies to the remaining candidate fields:
 - Mosaics: two resistant cultivars (RC1 and RC2, carrying the first and the second resistance sources, respectively) are assigned to candidate fields by re-running the allocation algorithm;
 - Mixtures: both RC1 and RC2 are allocated to all candidate fields;
 - Rotations: RC1 and RC2 are alternatively cultivated in candidate fields, depending on the number of cropping seasons over which a given cultivar is grown before being rotated;
 - Pyramiding: all candidate fields are cultivated with RC12, a resistant cultivar carrying both resistance sources.

Value

a shapefile containing the landscape structure (i.e. coordinates of field boundaries) and composition (i.e. cultivars) in time (i.e. each year) and space (i.e. each field).

References

Rimbaud L., Papaix J., Rey J.-F., Barrett L. G. and Thrall P. H. (2018). Assessing the durability and efficiency of landscape-based strategies to deploy plant resistance to pathogens. *PLoS Computational Biology* 14(4):e1006067.

Examples

```
## Generate a landscape consisting in a mosaic of fields cultivated with a susceptible cultivar
## and two resistant cultivars in balanced proportions and high level of spatial aggregation
## Not run:
landscapeTEST1
AgriLand(landscapeTEST1, filename="landscapeTEST1", propSR=2/3, iso1SR=3,
propRR=1/2, iso1RR=3, strat="MO", Nhost=3, nYears=30, seed=12345, graphic=TRUE)

## End(Not run)
```

demo_landsepi

Package demonstration

Description

Run a demonstration of the package.

Usage

```
demo_landsepi(seed = 12345)
```

Arguments

seed an integer used as seed value (for random number generator)

Details

Run demo landsepi

Run a 30-year simulated example of mosaic deployment strategy of two resistant cultivars in balanced proportions and high level of spatial aggregation. The generated model outputs are text files, graphics and a video.

dispH *Host dispersal data for testing.*

Description

A vectorised dispersal matrix of the host.

Usage

```
dispH
```

Format

The format is: num [1:24025] 1 0 0 0 0 0 0 0 0 ...

Details

In this example, the host is a cultivated plant: seeds are harvested and do not disperse. Thus the dispersal matrix is the identity matrix.

Examples

```
dispH  
summary(dispH)
```

dispP_1 *Pathogen dispersal data for testing.*

Description

Five vectorised dispersal matrices of the pathogen, associated with landscape 1, composed of 155 fields.

Usage

```
dispP_1
```

Format

The format is: num [1:24025] 8.81e-01 9.53e-04 7.08e-10 1.59e-10 3.29e-06 ...

Details

The pathogen dispersal matrix gives the probability for a pathogen in a field i (row) to migrate to field i' (column) through dispersal. It is computed based on a dispersal kernel and the euclidian distance between each point in fields i and i' , using the CaliFloPP algorithm (Bouvier et al. 2009). In this example, the dispersal kernel is an isotropic power-law function of equation: $f(x)=((b-2)*(b-1)/(2*\pi*a^2)) * (1 + x/a)^{-b}$ with $a=40$ a scale parameter and $b=7$ related to the weight of the dispersal tail. The expected mean dispersal distance is given by $2*a/(b-3)=20$ m.

References

Bouvier A, Ki  u K, Adamczyk K, Monod H. Computation of the integrated flow of particles between polygons. Environ. Model Softw. 2009;24(7):843-9. doi: <http://dx.doi.org/10.1016/j.envsoft.2008.11.006>.

Examples

```
dispP_1
summary(dispP_1)
## maybe str(dispP_1) ; plot(dispP_1) ...
```

dispP_2

Pathogen dispersal data for testing.

Description

Five vectorised dispersal matrices of the pathogen, associated with landscape 2, composed of 154 fields.

Usage

```
dispP_2
```

Format

The format is: num [1:23716] 8.70e-01 5.87e-08 3.35e-07 8.82e-10 2.55e-08 ...

Details

The pathogen dispersal matrix gives the probability for a pathogen in a field i (row) to migrate to field i' (column) through dispersal. It is computed based on a dispersal kernel and the euclidian distance between each point in fields i and i' , using the CaliFloPP algorithm (Bouvier et al. 2009). In this example, the dispersal kernel is an isotropic power-law function of equation: $f(x)=((b-2)*(b-1)/(2*\pi*a^2)) * (1 + x/a)^{-b}$ with $a=40$ a scale parameter and $b=7$ related to the weight of the dispersal tail. The expected mean dispersal distance is given by $2*a/(b-3)=20$ m.

References

Bouvier A, Ki  u K, Adamczyk K, Monod H. Computation of the integrated flow of particles between polygons. Environ. Model Softw. 2009;24(7):843-9. doi: <http://dx.doi.org/10.1016/j.envsoft.2008.11.006>.

Examples

```
dispP_2
summary(dispP_2)
## maybe str(dispP_2) ; plot(dispP_2) ...
```

dispP_3 *Pathogen dispersal data for testing.*

Description

Five vectorised dispersal matrices of the pathogen, associated with landscape 3, composed of 152 fields.

Usage

```
dispP_3
```

Format

The format is: num [1:23104] 8.35e-01 1.36e-09 1.85e-06 2.99e-09 5.11e-08 ...

Details

The pathogen dispersal matrix gives the probability for a pathogen in a field i (row) to migrate to field i' (column) through dispersal. It is computed based on a dispersal kernel and the euclidian distance between each point in fields i and i' , using the CaliFloPP algorithm (Bouvier et al. 2009). In this example, the dispersal kernel is an isotropic power-law function of equation: $f(x) = ((b-2)*(b-1)/(2*\pi*a^2)) * (1 + x/a)^{-b}$ with $a=40$ a scale parameter and $b=7$ related to the weight of the dispersal tail. The expected mean dispersal distance is given by $2*a/(b-3)=20$ m.

References

Bouvier A, Ki  u K, Adamczyk K, Monod H. Computation of the integrated flow of particles between polygons. Environ. Model Softw. 2009;24(7):843-9. doi: <http://dx.doi.org/10.1016/j.envsoft.2008.11.006>.

Examples

```
dispP_3
summary(dispP_3)
## maybe str(dispP_3) ; plot(dispP_3) ...
```

dispP_4 *Pathogen dispersal data for testing.*

Description

Five vectorised dispersal matrices of the pathogen, associated with landscape 4, composed of 153 fields.

Usage

```
dispP_4
```

Format

The format is: num [1:23409] 9.02e-01 3.02e-02 6.60e-11 1.12e-08 1.20e-08 ...

Details

The pathogen dispersal matrix gives the probability for a pathogen in a field i (row) to migrate to field i' (column) through dispersal. It is computed based on a dispersal kernel and the euclidian distance between each point in fields i and i' , using the CaliFloPP algorithm (Bouvier et al. 2009). In this example, the dispersal kernel is an isotropic power-law function of equation: $f(x) = ((b-2)*(b-1)/(2*\pi*a^2)) * (1 + x/a)^{-b}$ with $a=40$ a scale parameter and $b=7$ related to the weight of the dispersal tail. The expected mean dispersal distance is given by $2*a/(b-3)=20$ m.

References

Bouvier A, Ki  u K, Adamczyk K, Monod H. Computation of the integrated flow of particles between polygons. Environ. Model Softw. 2009;24(7):843-9. doi: <http://dx.doi.org/10.1016/j.envsoft.2008.11.006>.

Examples

```
dispP_4
summary(dispP_4)
## maybe str(dispP_4) ; plot(dispP_4) ...
```

dispP_5 *Pathogen dispersal data for testing.*

Description

Five vectorised dispersal matrices of the pathogen, associated with landscape 5, composed of 156 fields.

Usage

```
dispP_5
```

Format

The format is: num [1:24336] 8.82e-01 9.04e-10 2.32e-10 6.44e-07 3.48e-09 ...

Details

The pathogen dispersal matrix gives the probability for a pathogen in a field i (row) to migrate to field i' (column) through dispersal. It is computed based on a dispersal kernel and the euclidian distance between each point in fields i and i' , using the CaliFloPP algorithm (Bouvier et al. 2009). In this example, the dispersal kernel is an isotropic power-law function of equation: $f(x) = \frac{(b-2)^*(b-1)}{(2*\pi*a^2)} * (1 + x/a)^{-b}$ with $a=40$ a scale parameter and $b=7$ related to the weight of the dispersal tail. The expected mean dispersal distance is given by $2*a/(b-3)=20$ m.

References

Bouvier A, Ki  u K, Adamczyk K, Monod H. Computation of the integrated flow of particles between polygons. Environ. Model Softw. 2009;24(7):843-9. doi: <http://dx.doi.org/10.1016/j.envsoft.2008.11.006>.

Examples

```
dispP_5
summary(dispP_5)
## maybe str(dispP_5) ; plot(dispP_5) ...
```

HLIRdynamics

Generation of model outputs

Description

Generate epidemiological and evolutionary outputs from model simulations.

Usage

```
HLIRdynamics(pathRES, timeP, landscape, hostP, evolP, ecoP, graphic,
  video = FALSE, nMapPY = 5, th_break = 50000)
```

Arguments

pathRES	a character string indicating the path of the repository where outputs will be generated.
timeP	a list of simulation parameters (number of years, number of time-steps per year).
landscape	a shapefile containing the agricultural landscape (can be generated through function AgriLand).
hostP	a list of host parameters (number of cultivars, initial planting density, maximal carrying capacity, growth rate, reproduction rate, death rate, number of possible resistance sources (8), resistance formula, parameters of the sigmoid invasion function: kappa, sigma and s).

evolP	a list of evolution parameters (cost of infectivity, cost of aggressiveness, mutation rate, efficiency of major resistance genes, efficiency of quantitative resistance, trade-off strength, number of increments of quantitative resistance erosion, adaptation formula).
ecoP	a list of economic parameters (yield, purchase price, selling price).
graphic	a logical indicating if graphics of the outputs must be generated (TRUE) or not (FALSE).
video	a logical indicating if a video must be generated (TRUE) or not (FALSE). Works only if graphic is TRUE as well. Note that ffmpeg is required to generate videos.
nMapPY	an integer specifying the number of epidemic maps per year in the video (video must be TRUE).
th_break	an integer giving the threshold (number of infections) above which mutant pathogen are unlikely to go extinct, used to characterise resistance breakdown.

Details

Evolutionary outputs. *Durability of qualitative resistance:* For a given major gene, several computations are performed:

- (d1) time to first appearance of a pathogen mutant;
- (d2) time to first true infection of a resistant host by such mutants; and
- (d3) time when the number of infections of resistant hosts by these mutants reaches a threshold above which mutant pathogens are unlikely to go extinct.

Erosion of quantitative resistance: pathogen adaptation to quantitative resistance is gradual, so the three measures described above are computed for every step towards complete erosion of resistance (i.e. $nAgw-1$ levels).

Durability of a deployment strategy: a simulation run is divided into three periods:

1. the initial short-term period when all resistance sources are at their highest potential;
2. a transitory period during which a given deployment strategy is only partially effective; and
3. a longer-term period when all the resistances have been overcome or completely eroded.

To assess the end of the short-term period, the time to establishment (durability measure (d3)) is computed for every major gene, and every quantitative trait at the first level of erosion ($agw(p)=2$). The minimal value of these measures, denoted by D1, delimitates short-term and transitory periods. Similarly, the time to establishment is computed for every major gene, and for every quantitative trait at the highest level of erosion ($agw(p)=nAgw$). The maximal value of these measures, termed D2, delimits transitory and long-term periods.

Epidemiological outputs. The epidemiological impact of pathogen spread is evaluated by two different measures:

1. Green Leaf Area (GLA): The GLA represents the average number of productive hosts per time step and per surface unit.
2. Area Under Disease Progress Curve (AUDPC): The AUDPC is the average proportion of diseased hosts relative to the carrying capacity and represents disease severity.

Global epidemiological control: The GLA and AUDPC of every cultivar as well as the whole landscape are averaged across the whole simulation run, to measure the global epidemiological performance of a deployment strategy.

Short-term epidemiological control: The average GLA and AUDPC of the susceptible cultivar is computed on whole cropping seasons from the beginning of the simulation until the end of the season preceding year before D1.

Epidemiological control during the transitory period: The average GLA and AUDPC of the susceptible cultivar is computed on whole seasons from the beginning of the season following year after D1 to the end of the season year before preceding D2.

Long-term epidemiological control: The average GLA and AUDPC of the whole landscape is computed on whole seasons from the beginning of the year after D2 to the end of the simulation.

Value

A set of text files containing all outputs of the simulations (see details). A set of graphics and a video with epidemic maps can also be generated.

References

Rimbaud L., Papaïx J., Rey J.-F., Barrett L. G. and Thrall P. H. (2018). Assessing the durability and efficiency of landscape-based strategies to deploy plant resistance to pathogens. *PLoS Computational Biology* 14(4):e1006067.

Examples

```
demo_landsepi()
```

invlogit	<i>Inverse logit function</i>
----------	-------------------------------

Description

Given a numeric object return the invlogit of the values. Missing values (NAs) are allowed.

Usage

```
invlogit(x)
```

Arguments

x a numeric object

Details

The invlogit is defined by $\exp(x) / (1 + \exp(x))$. Values in x of -Inf or Inf return invlogits of 0 or 1 respectively. Any NAs in the input will also be NAs in the output.

Value

An object of the same type as x containing the invlogits of the input values.

Examples

```
invlogit(10)
```

landscapeTEST1	<i>A landscape for testing</i>
----------------	--------------------------------

Description

A landscape SpatialPolygons for testing, composed of 155 fields.

Usage

```
landscapeTEST1
```

Format

The format is a formal class 'SpatialPolygons' [package "sp"]

Details

The landscape structure is simulated using a T-tessellation algorithm (Kiêu et al. 2013) in order to control specific features such as number, area and shape of the fields.

References

Kiêu K, Adamczyk-Chauvat K, Monod H, Stoica RS. A completely random T-tessellation model and Gibbsian extensions. Spat. Stat. 2013;6:118-38. doi: <http://dx.doi.org/10.1016/j.spasta.2013.09.003>.

Examples

```
library(sp)
library(landsepi)
landscapeTEST1
plot(landscapeTEST1)
```

landscapeTEST2 *A landscape for testing*

Description

A landscape SpatialPolygons for testing, composed of 154 fields.

Usage

```
landscapeTEST2
```

Format

The format is a formal class 'SpatialPolygons' [package "sp"]

Details

The landscape structure is simulated using a T-tessellation algorithm (Kiêu et al. 2013) in order to control specific features such as number, area and shape of the fields.

References

Kiêu K, Adamczyk-Chauvat K, Monod H, Stoica RS. A completely random T-tessellation model and Gibbsian extensions. Spat. Stat. 2013;6:118-38. doi: <http://dx.doi.org/10.1016/j.spasta.2013.09.003>.

Examples

```
library(sp)
library(landsepi)
landscapeTEST2
plot(landscapeTEST2)
```

landscapeTEST3 *A landscape for testing*

Description

A landscape SpatialPolygons for testing, composed of 152 fields.

Usage

```
landscapeTEST3
```

Format

The format is formal class 'SpatialPolygons' [package "sp"].

Details

The landscape structure is simulated using a T-tessellation algorithm (Kiêu et al. 2013) in order to control specific features such as number, area and shape of the fields.

References

Kiêu K, Adamczyk-Chauvat K, Monod H, Stoica RS. A completely random T-tessellation model and Gibbsian extensions. *Spat. Stat.* 2013;6:118-38. doi: <http://dx.doi.org/10.1016/j.spasta.2013.09.003>.

Examples

```
library(sp)
library(landsepi)
landscapeTEST3
plot(landscapeTEST3)
```

landscapeTEST4	<i>A landscape for testing</i>
----------------	--------------------------------

Description

A landscape SpatialPolygons for testing, composed of 153 fields.

Usage

```
landscapeTEST4
```

Format

The format is formal class 'SpatialPolygons' [package "sp"]

Details

The landscape structure is simulated using a T-tessellation algorithm (Kiêu et al. 2013) in order to control specific features such as number, area and shape of the fields.

References

Kiêu K, Adamczyk-Chauvat K, Monod H, Stoica RS. A completely random T-tessellation model and Gibbsian extensions. *Spat. Stat.* 2013;6:118-38. doi: <http://dx.doi.org/10.1016/j.spasta.2013.09.003>.

Examples

```
library(sp)
library(landsepi)
landscapeTEST4
plot(landscapeTEST4)
```

landscapeTEST5	<i>A landscape for testing</i>
----------------	--------------------------------

Description

A landscape SpatialPolygons for testing, composed of 156 fields.

Usage

```
landscapeTEST5
```

Format

The format is formal class 'SpatialPolygons' [package "sp"]

Details

The landscape structure is simulated using a T-tessellation algorithm (Kiêu et al. 2013) in order to control specific features such as number, area and shape of the fields.

References

Kiêu K, Adamczyk-Chauvat K, Monod H, Stoica RS. A completely random T-tessellation model and Gibbsian extensions. *Spat. Stat.* 2013;6:118-38. doi: <http://dx.doi.org/10.1016/j.spasta.2013.09.003>.

Examples

```
library(sp)
library(landsepi)
landscapeTEST5
plot(landscapeTEST5)
```

logit	<i>Logit function</i>
-------	-----------------------

Description

Given a numeric object return the logit of the values. Missing values (NAs) are allowed.

Usage

```
logit(x)
```

Arguments

x a numeric object containing values between 0 and 1

Details

The logit is defined by $\log(x/(1-x))$. Values in x of 0 or 1 return logits of $-\text{Inf}$ or Inf respectively. Any NAs in the input will also be NAs in the output.

Value

An object of the same type as x containing the logits of the input values.

Examples

```
logit(0.5)
```

 model_landsepi

Model Landscape Epidemiology & Evolution

Description

Stochastic, spatially-explicit, demo-genetic model simulating the spread and evolution of a pathogen in a heterogeneous landscape.

Usage

```
model_landsepi(timeP, landscape, dispersal, inits, val_seed, hostP, pathoP,
  evolP)
```

Arguments

timeP	list of simulation parameters (number of years, number of time-steps per year)
landscape	landscape generated through AgriLand
dispersal	list of dispersal parameters (vectorised dispersal matrix of the pathogen, vectorised dispersal matrix of the host)
inits	list initial conditions (initial probability of infection by the pathogen)
val_seed	seed (for random number generation)
hostP	list of host parameters (number of cultivars, initial planting density, maximal carrying capacity, growth rate, reproduction rate, death rate, resistance formula, parameters of the sigmoid invasion function: kappa, sigma and s)
pathoP	list of pathogen parameters (probability to survive the off-season, probability to reproduce via sex rather than via cloning, infection rate, reproduction rate, average latent period duration, variance of the latent period, average infectious period duration, variance of the infectious period duration, parameters of the sigmoid contamination function: kappa, sigma, s)
evolP	list of evolution parameters (cost of infectivity, cost of aggressiveness, mutation rate, efficiency of major resistance genes, efficiency of quantitative resistance, trade-off strength, number of increments of quantitative resistance erosion, average time to expression of quantitative resistance, Variance of the time to expression of quantitative resistance, adaptation formula)

Details

- The model is stochastic, spatially explicit (the basic spatial unit is an individual field), based on a SEIR ('susceptible-exposed-infectious-removed') structure with a discrete time step. It simulates the spread and evolution of a pathogen in an agricultural landscape, across cropping seasons split by host harvests which impose potential bottlenecks to the pathogen.
- A wide array of deployment strategies can be simulated: mosaics, mixtures, rotations and pyramiding of multiple major resistance genes which affect pathogen infectivity, and up to four quantitative resistance traits. These traits target different aggressiveness components of the pathogen, i.e. the infection rate, the duration of the latent period and the infectious period, and the propagule production rate. Quantitative resistance may be expressed from the time of planting, or later in the cropping season (Adult Plant Resistance or Mature Plant Resistance).
- The genotype of cultivated plant cultivars is specified using the "resistance formulas", i.e. a vector of size 8. the four first elements indicate whether the cultivar carries major resistance genes #1, #2, #3 and #4, respectively. The following four elements indicate whether the cultivar carried a quantitative resistance trait against the infection rate, the latent period duration, the sporulation rate, or the sporulation duration of the pathogen, respectively. For example, the formula $c(1,0,0,0,0,1,0,0)$ indicates the presence of major gene #1 and a quantitative resistance which increases the duration of the latent period of the pathogen.
- Initially, the pathogen is not adapted to any source of resistance, and is only present on susceptible hosts. However, through mutation, it can evolve and may acquire infectivity genes (which leads to breakdown of major resistance genes) or increase aggressiveness (which leads to the erosion of the relevant quantitative resistance traits). These genes may also be reassorted via sexual reproduction.
- Evolution of a pathogen toward infectivity or increased aggressiveness on a resistant host is often penalised by a fitness cost on susceptible hosts. Consequently, in the present model, pathogens carrying infectivity genes may have reduced infection rate (cost of infectivity) on susceptible hosts relative to pathogens that do not carry these genes. Similarly, a gain in pathogen aggressiveness on quantitatively resistant hosts is penalised by a decreased aggressiveness on susceptible hosts, leading to a trade-off.

Value

A set of binary files is generated for every year of simulation and every compartment:

- H: healthy hosts,
- Hjuv: juvenile healthy hosts,
- L: latently infected hosts,
- I: infectious hosts,
- R: removed hosts,
- S: propagules.

Each file indicates for every time-step the number of individuals in each field, and when appropriate for each cultivar and pathotype)

References

Rimbaud L., Papaix J., Rey J.-F., Barrett L. G. and Thrall P. H. (2018). Assessing the durability and efficiency of landscape-based strategies to deploy plant resistance to pathogens. *PLoS Computational Biology* 14(4):e1006067.

multiN

Allocation of cultivars

Description

Algorithm based on latent Gaussian fields to allocate two different types of crops across the simulated landscapes.

Usage

```
multiN(d, area, range, prop)
```

Arguments

d	a matrix of the pairwise distances between the centroids of the fields of the landscape.
area	vector containing the areas of the fields.
range	range of spatial autocorrelation between fields (<0 for fragmented landscapes, >0 for aggregated landscapes, =0 for random allocation).
prop	relative proportion of the second type of crop.

Details

This algorithm allows the control of the proportions of each type of crop in terms of surface coverage, and their level of spatial aggregation. A random vector of values is drawn from a multivariate normal distribution with expectation 0 and a variance-covariance matrix which depends on the pairwise distances between the centroids of the fields. The variance-covariance matrix is computed from a periodic function for highly fragmented or highly aggregated types of crops, an exponential function for moderately aggregated types of crops, and from a normal distribution for a random allocation of the types of crops. Next, the two types of crops are allocated to different fields depending on whether the each value drawn from the multivariate normal distribution is above or below a threshold. The proportion of each type of crop in the landscape is controlled by the value of this threshold (parameter prop).

Value

A dataframe containing the index of each field (column 1) and the index (0 or 1) of the type of crop grown on these fields (column 2).

Examples

```
## Not run:
d <- matrix(rpois(100,100), nrow=10)
area <- data.frame(num=1:10, area=10)
## Fragmented landscape
multiN(d, area, range=-2, prop=0.5)

## End(Not run)
```

periodic_cov	<i>Periodic covariance function</i>
--------------	-------------------------------------

Description

Periodic function used to compute the variance-covariance matrix of the fields of the landscape.

Usage

```
periodic_cov(d, range, phi = 1)
```

Arguments

d	a numeric object containing pairwise distances between the centroids of the fields
range	range (half-period of oscillations)
phi	amplitude of the oscillations

Details

The periodic covariance is defined by $\exp(-2 * \sin(d*\pi/(2*range))^2 / \phi^2)$. It is used to generate highly fragmented or highly aggregated landscapes.

Value

An object of the same type as d.

Examples

```
periodic_cov(10, range=5)
```

plotevolQR *Plotting pathotype frequencies*

Description

Plot the dynamic of pathotype frequencies in a tiff file.

Usage

```
plotevolQR(pathRES, nIncr, trait, I_aggrProp, D, nTS, nYears, nTSpY)
```

Arguments

pathRES	the path of the repository where the graphic will be generated
nIncr	number of pathotypes
trait	index of the evolving trait (1 to 4 for major genes, 5 to 8 for quantitative resistance traits against infection rate (5), latent period (6), reproduction rate (7) and infectious period duration (8))
I_aggrProp	a matrix giving the frequency of every pathotype (row) for every time-step (columns)
D	durability of the trait
nTS	number of simulated time-steps
nYears	number of simulated years
nTSpY	number of time-steps per year

Examples

```
## Not run:
freqMatrix <- matrix(0, nrow=2, ncol=100)
freqMatrix[2,26:100] <- (26:100)/100
freqMatrix[1,] <- 1-freqMatrix[2,]
plotevolQR(getwd(), nIncr=2, trait=1, freqMatrix, D=25, nTS=100,nYears=10,nTSpY=10)

## End(Not run)
```

plotland *Plotting the landscape*

Description

Plot a landscape with colors or hatched lines to represent different types of fields

Usage

```
plotland(landscape, COL = rep(0, length(landscape)), DENS = rep(0,
  length(landscape)), ANGLE = rep(30, length(landscape)),
  COL.LEG = unique(COL), DENS.LEG = unique(DENS),
  ANGLE.LEG = unique(ANGLE), TITLE = "", SUBTITLE = "",
  LEGEND1 = rep("", length(COL.LEG)), LEGEND2 = rep("",
  length(COL.LEG)), TITLE.LEG2 = "", XMAX = 2000, YMAX = 2000)
```

Arguments

landscape	a spatialpolygon object containing field coordinates
COL	vector containing the color of each field
DENS	vector containing the density of hatched lines for each field
ANGLE	vector containing the angle of hatched lines for each field
COL.LEG	vector containing the colors in the first legend
DENS.LEG	vector containing the density of hatched lines in the second legend
ANGLE.LEG	vector containing the angle of hatched lines in the second legend
TITLE	title of the graphic
SUBTITLE	subtitle of the graphic
LEGEND1	labels in the first legend (colors)
LEGEND2	labels in the second legend (hatched lines)
TITLE.LEG2	title for the second legend
XMAX	maximal coordinate on horizontal axis
YMAX	maximal coordinate on vertical axis

Examples

```
## Not run:
## Draw a landscape with various colours
landscapeTEST1
plotland(landscapeTEST1, COL=1:length(landscapeTEST1),
  DENS=rep(0,length(landscapeTEST1)), ANGLE=rep(30,length(landscapeTEST1)))

## End(Not run)
```

 simul_landsepi

Simulation with provided data

Description

Simulation of plant resistance deployment, using landscape structures provided with the package and a parameterisation of the model to represent pathogens as typified by rusts of cereals (e.g. stripe rust, stem rust, and leaf rust of wheat and barley). All parameters are optional. See details for explanations.

Usage

```
simul_landsepi(seed = 12345, nYears = 5, nTSpY = 120, idLan = 1,
  propSR = 2/3, isoLSR = 1, propRR = 1/2, isoLRR = 1,
  strat = "MO", Nhost = 3, pI0 = 5e-04, resistance1 = c(1, 0, 0, 0,
  0, 0, 0, 0), resistance2 = c(0, 1, 0, 0, 0, 0, 0, 0),
  costInfect = 0.75, costAggr = 0.75, taumut = 1e-07, probSex = 0,
  MGeff = 1, QReff = 0.5, beta = 1, Naggr = 6, timeToQR_exp = 0,
  timeToQR_var = 0, C0 = rep(0.1, Nhost), Cmax = rep(2, Nhost),
  yield = cbind(H = rep(1/(120 * 1000), Nhost), L = rep(0, Nhost), I =
  rep(0, Nhost), R = rep(0, Nhost)), purchPrice = rep(0.001, Nhost),
  sellPrice = rep(0.2, Nhost), graphic = TRUE, video = FALSE)
```

Arguments

seed	an integer used as seed value (for random number generator).
nYears	an integer giving the number of simulated years.
nTSpY	an integer giving the number of time steps per year.
idLan	an integer giving the index of landscape structure (1 to 5).
propSR	proportion of fields where resistance is deployed: $(RC)/(SC+RC)$ or $(RC1+RC2)/(SC+RC1+RC2)$. Must be between 0 and 1.
isoLSR	an integer giving the spatial aggregation of fields where resistance is deployed (1=highly fragmented, 2=balanced, 3=highly aggregated).
propRR	when applicable (mixtures and mosaics only), relative proportion of the second resistant cultivar: $(RC2)/(RC1+RC2)$. Must be between 0 and 1.
isoLRR	when applicable, an integer specifying the spatial (for mosaics: 1=highly fragmented, 2=balanced, 3=highly aggregated) or temporal (for rotations: 1=every year, 2=every two years, 3=every three years) aggregation of fields cultivated with the second resistant cultivar.
strat	a character string specifying the deployment strategy ("MO"=mosaic, "MI"=mixture, "RO"=rotations, "PY"=pyramiding).
Nhost	an integer giving the number of cultivars (1, 2 or 3).
pI0	initial probability of infection of the susceptible cultivar. Must be between 0 and 1.
resistance1	a logical vector of size 8 giving the resistance formula of the 2nd cultivar (see details)
resistance2	when applicable, a logical vector of size 8 giving the resistance formula of the 3rd cultivar (see details)
costInfect	cost of infectivity paid by infective pathogens (i.e. adapted to plant cultivars carrying a major gene) on susceptible hosts. Must be between 0 and 1.
costAggr	cost of aggressiveness paid by fully adapted pathogens (relative to plant cultivars carrying a quantitative resistance trait) on fully susceptible hosts. Must be between 0 and 1.
taumut	mutation probability: probability for a propagule to change its infectivity or its aggressiveness on a resistant cultivar carrying a major gene or a quantitative resistance trait. Must be above 0. If equal to 0, then the pathogen cannot evolve.

probSex	probability for an infection that its reproduction is sexual rather than clonal.
MGeff	efficiency of major-gene resistance on the infection rate of non-adapted pathogens. Must be between 0 and 1.
QReff	efficiency of quantitative resistance on the target aggressiveness trait (infection rate, latent period duration, sporulation rate, or sporulation duration) of non-adapted pathogens. Must be between 0 and 1.
beta	trade-off strength for pathogen adaptation to quantitative resistance (<1 for weak, =1 for linear, >1 for strong). Must be above 0.
Naggr	an integer specifying the number of increments to completely adapt to quantitative resistance. Must be greater or equal 2.
timeToQR_exp	average time to expression of quantitative resistance (to simulate Adult Plant Resistance).
timeToQR_var	variance of the time to expression of quantitative resistance (to simulate Adult Plant Resistance).
C0	planting density of the different cultivars (in number of hosts per meter square).
Cmax	carrying capacity of the different cultivars (in number of hosts per meter square).
yield	a matrix of yield (weight units of product/individual/time-step) for each cultivar (rows) and each sanitary state (columns: H, L, I, R).
purchPrice	price of crop planting (in monetary units/planted individual)
sellPrice	selling price (in monetary unit/weight units of production)
graphic	a logical indicating if graphics must be generated (TRUE) or not (FALSE).
video	a logical indicating if a video must be generated (TRUE) or not (FALSE). Works only if graphic is TRUE as well.

Details

Landscape structure The landscape structure is the physical structure of the area, defined as the spatial arrangement of fields.

Deployment strategies Deployment strategies include the deployment of a susceptible cultivar (SC) and one (RC) or two (RC1 and RC2) resistant cultivars carrying up to four major resistance genes or up to four quantitative resistance traits (against infection rate, latent period, sporulation rate and sporulation duration of the pathogen). In addition, the different resistance sources can be combined in time (crop rotation: recurrent succession of cultivars in the same field), or space within a single cultivar (pyramiding), in different cultivars of the same field (mixtures) or in different fields (mosaics).

Resistance formulas The genetic resistance carried by a plant cultivar is specified by a vector of size 8: the four first elements indicate whether the cultivar carries major resistance genes #1, #2, #3 and #4, respectively. The following four elements indicate whether the cultivar carried a quantitative resistance trait against the infection rate, the latent period duration, the sporulation rate, or the sporulation duration of the pathogen, respectively. For example, the formula $c(1,0,0,0,0,1,0,0)$ indicates the presence of major gene #1 and a quantitative resistance which increases the duration of the latent period of the pathogen.

Model outputs **Evolutionary outputs.** *Durability of qualitative resistance:* For a given major gene, several computations are performed:

- (d1) time to first appearance of a pathogen mutant;
- (d2) time to first true infection of a resistant host by such mutants; and
- (d3) time when the number of infections of resistant hosts by these mutants reaches a threshold above which mutant pathogens are unlikely to go extinct.

Erosion of quantitative resistance: pathogen adaptation to quantitative resistance is gradual, so the three measures described above are computed for every step towards complete erosion of resistance (i.e. $nAgw-1$ levels).

Durability of a deployment strategy: a simulation run is divided into three periods:

1. the initial short-term period when all resistance sources are at their highest potential;
2. a transitory period during which a given deployment strategy is only partially effective; and
3. a longer-term period when all the resistances have been overcome or completely eroded.

To assess the end of the short-term period, the time to establishment (durability measure (d3)) is computed for every major gene, and every quantitative trait at the first level of erosion ($agw(p)=2$). The minimal value of these measures, denoted by D1, delimitates short-term and transitory periods. Similarly, the time to establishment is computed for every major gene, and for every quantitative trait at the highest level of erosion ($agw(p)=nAgw$). The maximal value of these measures, termed D2, delimits transitory and long-term periods.

Epidemiological outputs. The epidemiological impact of pathogen spread is evaluated by two different measures:

1. Green Leaf Area (GLA): The GLA represents the average number of productive hosts per time step and per surface unit.
2. Area Under Disease Progress Curve (AUDPC): The AUDPC is the average proportion of diseased hosts relative to the carrying capacity and represents disease severity.

Global epidemiological control: The GLA and AUDPC of every cultivar as well as the whole landscape are averaged across the whole simulation run, to measure the global epidemiological performance of a deployment strategy.

Short-term epidemiological control: The average GLA and AUDPC of the susceptible cultivar is computed on whole cropping seasons from the beginning of the simulation until the end of the season preceding year before D1.

Epidemiological control during the transitory period: The average GLA and AUDPC of the susceptible cultivar is computed on whole seasons from the beginning of the season following year after D1 to the end of the season year before preceding D2.

Long-term epidemiological control: The average GLA and AUDPC of the whole landscape is computed on whole seasons from the beginning of the year after D2 to the end of the simulation.

Value

A set of binary files is generated for every year of simulation and every compartment:

- H: healthy hosts,
- Hjuv: juvenile healthy hosts,

- L: latently infected hosts,
- I: infectious hosts,
- R: removed hosts,
- S: propagules.

Each file indicates for every time-step the number of individuals in each field, and when appropriate for each cultivar and pathotype) These binary files are used to generate a set of text files containing all outputs of the simulations (see details). A set of graphics and a video showing epidemic maps can also be generated.

References

Rimbaud L., Papaix J., Rey J.-F., Barrett L. G. and Thrall P. H. (2018). Assessing the durability and efficiency of landscape-based strategies to deploy plant resistance to pathogens. *PLoS Computational Biology* 14(4):e1006067.

Examples

```
## Default parameterisation (5-year simulation of a mosaic deployment strategy of
## two resistant cultivars in balanced proportions and low level of spatial aggregation)
simul_landsepi()

## Mosaic of two major genes
simul_landsepi(seed=1, idLan=1, propSR=2/3, isolSR=3, propRR=1/2, isolRR=3, strat="M0", Nhost=3
, nYears=50, resistance1=c(1,0,0,0,0,0,0,0), resistance2=c(0,1,0,0,0,0,0,0)
, costInfect=0.5, taumut=1e-7)

## Mixture of two major genes
simul_landsepi(seed=1, idLan=1, propSR=2/3, isolSR=3, propRR=1/2, strat="MI", Nhost=3
, nYears=50, resistance1=c(1,0,0,0,0,0,0,0), resistance2=c(0,1,0,0,0,0,0,0)
, costInfect=0.5, taumut=1e-7)

## Rotations of two major genes
simul_landsepi(seed=1, idLan=1, propSR=2/3, isolSR=3, isolRR=1, strat="R0", Nhost=3
, nYears=50, resistance1=c(1,0,0,0,0,0,0,0), resistance2=c(0,1,0,0,0,0,0,0)
, costInfect=0.5, taumut=1e-7)

## Pyramiding of two major genes
simul_landsepi(seed=1, idLan=1, propSR=2/3, isolSR=3, strat="PY", Nhost=2
, nYears=50, resistance1=c(1,1,0,0,0,0,0,0), costInfect=0.5, taumut=1e-7)

## Combination of a major gene with a quantitative resistance against the latent period
simul_landsepi(seed=1, idLan=1, propSR=0.8, isolSR=1, strat="PY", Nhost=2
, nYears=50, resistance1=c(1,0,0,0,0,1,0,0)
, costInfect=0.5, costAggr=0.5, taumut=1e-7, MGeff=1.0, QReff=0.5, beta=1.0, Naggr=6)
```

Index

- *Topic **SEIR**
 - landsepi-package, 2
- *Topic **datasets**
 - dispH, 8
 - dispP_1, 8
 - dispP_2, 9
 - dispP_3, 10
 - dispP_4, 11
 - dispP_5, 11
 - landscapeTEST1, 15
 - landscapeTEST2, 16
 - landscapeTEST3, 16
 - landscapeTEST4, 17
 - landscapeTEST5, 18
- *Topic **demo-genetic**
 - landsepi-package, 2
- *Topic **deployment**
 - landsepi-package, 2
- *Topic **durability**
 - landsepi-package, 2
- *Topic **model**
 - landsepi-package, 2
- *Topic **resistance**
 - landsepi-package, 2
- *Topic **spatial**
 - landsepi-package, 2
- *Topic **stochastic**
 - landsepi-package, 2
- _PACKAGE (landsepi-package), 2

AgriLand, 5

- demo_landsepi, 7
- dispH, 8
- dispP_1, 8
- dispP_2, 9
- dispP_3, 10
- dispP_4, 11
- dispP_5, 11

- HLIRdynamics, 12
- invlogit, 14
- landscapeTEST1, 15
- landscapeTEST2, 16
- landscapeTEST3, 16
- landscapeTEST4, 17
- landscapeTEST5, 18
- landsepi (landsepi-package), 2
- landsepi-package, 2
- logit, 18
- model_landsepi, 19
- multiN, 21
- periodic_cov, 22
- plotevolQR, 23
- plotland, 23
- simul_landsepi, 24